A Scalable Gradient-Based Optimization Framework for Sparse Minimum-Variance Portfolio Selection

Sarat Moka* School of Mathematics and Statistics University of New South Wales Sydney, NSW, Australia

Vali Asimit Bayes Business School City St George's, University of London London, United Kingdom

Matias Quiroz School of Mathematical and Physical Sciences University of Technology Sydney Sydney, NSW, Australia

> Samuel Muller Faculty of Science and Engineering Macquarie University Sydney, NSW, Australia

Abstract

Portfolio optimization involves selecting asset weights to minimize a risk-reward objective, such as the portfolio variance in the classical minimum-variance framework. Sparse portfolio selection extends this by imposing a cardinality constraint: only k assets from a universe of p may be included. The standard approach models this problem as a mixed-integer quadratic program and relies on commercial solvers to find the optimal solution. However, the computational costs of such methods increase exponentially with k and p, making them too slow for problems of even moderate size. We propose a fast and scalable gradient-based approach that transforms the combinatorial sparse selection problem into a constrained continuous optimization task via Boolean relaxation, while preserving equivalence with the original problem on the set of binary points. Our algorithm employs a tunable parameter that transmutes the auxiliary objective from a convex to a concave function. This allows a stable convex starting point, followed by a controlled path toward a sparse binary solution as the tuning parameter increases and the objective moves toward concavity. In practice, our method matches commercial solvers in asset selection for most instances and, in rare instances, the solution differs by a few assets whilst showing a negligible error in portfolio variance.

1 Introduction

We propose a sparse portfolio selection framework that is computationally fast. To ground our ideas, we consider a specific instantiation of this framework—one designed for tractability and rigorous theoretical analysis. In particular, we consider the problem of minimizing portfolio variance across a universe of p assets under a sparsity constraint that restricts investments to at most k assets. Minimum-variance portfolio is a cornerstone of portfolio theory, providing a principled risk-reduction strategy by focusing solely on return volatility. Originating in the seminal mean-variance framework of Markowitz (1952), the minimum-variance portfolio remains widely adopted in practice (Clarke et al., 2011), especially when return forecasts are unreliable or difficult to estimate. Recent advances in portfolio selection—as discussed in Section 2.3—prioritize simplicity by enforcing sparsity (i.e.,

^{*}Corresponding author: s.moka@unsw.edu.au

limiting non-zero asset weights). Sparsity enhances interpretability, mitigates estimation error, and improves practical feasibility (Hastie et al., 2015; Brodie et al., 2009; Fan et al., 2012; DeMiguel et al., 2009). In contrast, conventional portfolio optimization often yields dense allocations, where investing in assets with negligible weights inflates transaction costs and amplify sensitivity to covariance matrix estimation errors (Gao and Li, 2013).

The sparse minimum-variance problem (see Section 2.2) is naturally framed as a mixed-integer quadratic program. Specialized commercial solvers—including branch-and-bound algorithms (Land and Doig, 1960) which guarantee globally optimal solutions given adequate computational time—exist for such problems. However, their runtimes scale exponentially with problem size and become infeasible in practical applications for moderately large p and k.

We recast the sparse minimum-variance portfolio problem as a binary-constrained optimization over the binary k-cube $\{s \in \{0,1\}^p : \sum_{j=1}^p s_j = k\}$. Drawing on the Boolean relaxation approach for best subset selection in regression by Moka et al. (2024), we introduce a continuous auxiliary objective function defined on the simplex $\{t \in [0,1]^p : \sum_{j=1}^p t_j = k\}$, which coincides with the reformulated function's values on the binary k-cube. Moreover, the auxiliary objective function has a positive tuning parameter that gradually transitions this auxiliary function from convex to concave as it increases. The convex phase facilitates stable initialization, while the concave phase ensures equivalence between the auxiliary function's minima and those of the original problem. Building on these properties, we develop a variant of the *Frank-Wolfe* (conditional gradient) method that iteratively increases the parameter during optimization, thereby guiding the objective from convexity to concavity to reach the optimal solution of the binary constrained problem.

From a theoretical standpoint, as the tuning parameter increases continuously, our method is guaranteed to converge to an optimal solution of the original problem with binary constraints. In practical implementations, where the parameter is discretized, solutions may marginally deviate from optimality. Nonetheless, extensive numerical experiments on synthetic and real-world datasets demonstrate that such deviations are limited to a small number of assets, with minimal error in the objective value. Importantly, our algorithm eliminates combinatorial complexity entirely, enabling a computationally efficient and scalable solution.

The remainder of the paper is organized as follows. Section 2 formulates the sparse portfolio optimization problem and reviews existing approaches. Section 3 introduces our Boolean relaxation framework, establishes its theoretical guarantees, and presents our algorithm. Section 4 demonstrates the performance of our method on synthetic and real-world financial datasets. Finally, Section 5 concludes the paper and outlines future research. All the theoretical results are proved in Appendix B.

2 Sparse portfolio optimization

In this section, we first introduce the notation employed in this paper and then present the formulation of the sparse portfolio selection problem. We use boldface notation for column vectors, for example, $u \in \mathbb{R}^p$ denotes a *p*-dimensional real vector. We use u^{\top} to denote the transpose of a vector u. The all-zeros vector, the all-ones vector, and the identity matrix are denoted by $\mathbf{0}$, $\mathbf{1}$, and I, respectively, and their dimensions are clear from the context. Capital letters denote matrices, unless otherwise mentioned. For a binary vector $s \in \{0, 1\}^p$ that denotes which assets are selected or not, $|s| = \sum_{j=1}^{p} s_j$ denotes the number of selected assets. Moreover, for a $p \times p$ -dimensional matrix A, $A_{[s]} \in \mathbb{R}^{|s| \times |s|}$ is the square sub-matrix of A obtained by removing all columns and rows of A for which $s_j = 0$, $j = 1, \ldots, p$. Similarly, the |s|-dimensional vector $u_{[s]}$ keeps the elements of the *p*-dimensional vector $u \in \mathbb{R}^p$ corresponding to $s_j = 1$. Throughout, two optimization problems are said to be equivalent if the solutions of one problem provides the solutions to the other, and vice versa.

2.1 Minimum-variance portfolio optimization

Let $\boldsymbol{x}_{\tau} = (x_{\tau,1}, \dots, x_{\tau,p})^{\top} \in \mathbb{R}^p$ denote the vector of returns for p assets at time τ . Consider a portfolio with weights $\boldsymbol{\beta} = (\beta_1, \dots, \beta_p)^{\top}$ such that the sum of the weights equals one, i.e., $\boldsymbol{\beta}^{\top} \mathbf{1} = \sum_{j=1}^p \beta_j = 1$. The portfolio return at time τ is $\boldsymbol{\beta}^{\top} \boldsymbol{x}_{\tau}$. The mean return of the portfolio is given by $\boldsymbol{\beta}^{\top} \boldsymbol{\mu}$, and its risk (volatility) is characterized by the variance $\boldsymbol{\beta}^{\top} \Sigma \boldsymbol{\beta}$, where $\boldsymbol{\mu}$ represents the mean vector of asset returns and Σ is the covariance matrix of \boldsymbol{x}_{τ} . The minimum-variance portfolio is an asset allocation strategy that constructs a portfolio with the lowest possible risk, measured as the variance of portfolio returns, regardless of expected returns. This strategy is ideal for risk-averse investors who prioritize minimizing risk over maximizing returns and is particularly useful when asset returns are difficult to forecast. The asset allocation for such an investor is given by the optimal weights obtained by solving

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^p} \boldsymbol{\beta}^\top \Sigma \boldsymbol{\beta}, \quad \text{subject to } \boldsymbol{\beta}^\top \mathbf{1} = 1.$$
 (1)

We make the following key assumption throughout the paper.

Assumption 1. The covariance matrix Σ is positive definite.

This assumption is not restrictive in practice because we can obtain Σ using a positive definite covariance matrix estimator. Moreover, in practice, problem (1) is often augmented with a ridge regularization term by replacing the covariance matrix Σ with $\Sigma + \lambda I$ for some $\lambda > 0$; see Fastrich et al. (2015); Roncalli (2013). In this case, Assumption 1 holds trivially. The following lemma gives the optimal weights (i.e., the unique solution to (1)) under this assumption.

Lemma 1. The solution to the minimum-variance problem (1) is

$$\widehat{\boldsymbol{\beta}} = \frac{\boldsymbol{\Sigma}^{-1} \mathbf{1}}{\mathbf{1}^{\top} \boldsymbol{\Sigma}^{-1} \mathbf{1}}.$$

2.2 Sparse portfolio selection

The minimum-variance optimization (1) yields dense portfolios, assigning nonzero weights to all assets. In many applications, some of the weights may be very small and, ideally, such assets should not be invested in to avoid transaction costs. Moreover, including only the relevant assets may reduce estimation errors. To mitigate these issues, *sparse portfolio selection* enforces an additional constraint on the number of assets selected, resulting in the optimization problem: For an integer $k \ge 1$,

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^p} \boldsymbol{\beta}^\top \Sigma \boldsymbol{\beta}, \quad \text{subject to } \mathbf{1}^\top \boldsymbol{\beta} = 1, \ \|\boldsymbol{\beta}\|_0 \le k,$$
(2)

where $\|\beta\|_0$ denotes the number of non-zero elements in β . This NP-hard combinatorial problem makes naive exhaustive searches—evaluating all possible subsets of k assets from p—computationally intractable even for moderately large k and p (Natarajan, 1995; Bertsimas et al., 2016).

A more efficient way to tackle problem (2) is via mixed-integer quadratic programming; for a comprehensive survey of its use in portfolio selection, refer to Mencarelli and d'Ambrosio (2019). A widely adopted technique in mixed-integer optimization is the *Big-M formulation*, which enforces logical conditions—such as whether asset j is included ($s_j = 1$) or excluded ($s_j = 0$)—by embedding them directly into linear constraints. In particular, the Big-M formulation of problem (2) is

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^{p}, \ \boldsymbol{s} \in \{0,1\}^{p}} \boldsymbol{\beta}^{\top} \Sigma \boldsymbol{\beta},$$

subject to $\mathbf{1}^{\top} \boldsymbol{\beta} = 1, \ \mathbf{1}^{\top} \boldsymbol{s} \leq k, \text{ and}$
 $-Ms_{j} \leq \beta_{j} \leq Ms_{j}, \ \forall j = 1, \dots, p,$ (3)

where M > 0 is such that at every optimal solution β^* of (2), $\max_j |\beta_j^*| \le M$. This formulation ensures that $\beta_j = 0$ if and only if $s_j = 0$.

Commercial solvers such as CPLEX (IBM Corporation, 2024) and Gurobi (Gurobi Optimization, LLC, 2024) employ branch-and-bound algorithms (Land and Doig, 1960) to tackle formulation (3). Although they are guaranteed to converge to the global optimum given sufficient time, their computational cost grows exponentially, rendering them impractical for even moderately large k and p.

2.3 Limitations and related work

As a first instantiation of our framework, we focus on a tractable setting that excludes additional weight constraints (e.g., short selling or minimum investments), as well as mean-variance objectives and risk-free assets. We leave these extensions for future work. Although we prove that our method recovers the exact sparse k optimal solution as the tuning parameter increases continuously, this guarantee

breaks down in practical implementations, where the tuning parameter is discretized and each step is only solved approximately using a gradient-based procedure. In contrast, certifiable optimization methods (e.g., the Big-M formulation) can guarantee optimality given enough computational budget. However, our method is computationally feasible for large-scale problems and achieves better solutions than the Big-M formulation under a fixed computational budget.

Bertsimas and coauthors (Bertsimas and Cory-Wright, 2022; Bertsimas et al., 2021) present a major advance in scalable, certifiably optimal sparse portfolio selection under a mean-variance objective with weight constraints (e.g., minimum investments, no short selling). They incorporate a ridge penalty to improve tractability, and their cutting-plane algorithm outperforms the Big-M formulation for the ridge-regularized problem while converging faster with stronger regularization. However, this reliance on a regularization term makes their approach less applicable to our penalty-free setting. Other important works in cardinality-constrained minimization is surveyed in Tillmann et al. (2024).

3 Methodology

To overcome the combinatorial bottleneck inherent in problem (2), we introduce a Boolean relaxation that replaces the binary constraints with continuous ones, yielding an auxiliary objective function whose gradient and Hessian can be computed in closed form. This enables scalable continuous optimization.

3.1 Boolean relaxation

The first step in our Boolean relaxation of the sparse portfolio selection problem (2) is to rewrite it as a binary constrained problem. It follows from Assumption 1 that each principal submatrix $\Sigma_{[s]}$ is invertible for every $s \neq 0$ (Bhatia, 2009). Consequently, (2) is equivalent to the binary constrained problem

$$\min_{\boldsymbol{s}\in\{0,1\}^{p}} \min_{\boldsymbol{\beta}_{[\boldsymbol{s}]}\in\mathbb{R}^{|\boldsymbol{s}|}} \boldsymbol{\beta}_{[\boldsymbol{s}]}^{\top} \boldsymbol{\Sigma}_{[\boldsymbol{s}]} \boldsymbol{\beta}_{[\boldsymbol{s}]}, \quad \text{subject to } \mathbf{1}^{\top} \boldsymbol{\beta} = 1, \ \mathbf{1}^{\top} \boldsymbol{s} \leq k,$$
(4)

where in β (without subscript), $\beta_j = 0$ for $s_j = 0$. Using the notation $\Sigma_{[s]}^{-1} = (\Sigma_{[s]})^{-1}$, from Lemma 1, $\hat{\beta}_{[s]} = \Sigma_{[s]}^{-1} \mathbf{1}/(\mathbf{1}^{\top} \Sigma_{[s]}^{-1} \mathbf{1})$ is the solution of the inner minimization in (4). Thus, the problem can be rewritten as

$$\min_{\boldsymbol{s}\in\{0,1\}^p} \frac{1}{\boldsymbol{1}^\top \boldsymbol{\Sigma}_{[\boldsymbol{s}]}^{-1} \boldsymbol{1}}, \quad \text{subject to } \boldsymbol{1}^\top \boldsymbol{s} \le k,$$
(5)

or, equivalently,

$$\min_{\boldsymbol{s}\in\{0,1\}^p} -\mathbf{1}^\top \Sigma_{[\boldsymbol{s}]}^{-1} \mathbf{1}, \quad \text{subject to } \mathbf{1}^\top \boldsymbol{s} \le k,$$
(6)

where we take $\mathbf{1}^{\top} \Sigma_{[s]}^{-1} \mathbf{1}$ to be zero when $s = \mathbf{0}$. Note that solving (6), which we refer to as the *target* problem, is equivalent to solving the original problem (2). Theorem 1 establishes the monotonicity of the optimal value of (6) as a function of k. In particular, it implies that the inequalities in (4), (5) and (6) can be replaced with equalities. In other words, inclusion of additional assets (increasing k) can only improve (lower) or retain the same variance of (2).

Theorem 1. The optimal value of the target problem (6) is non-increasing in $k \in \{0, 1, ..., p\}$.

We now provide a Boolean relaxation of (6) as an auxiliary continuous function on $[0, 1]^p$, controlled by a tuning parameter $\delta > 0$. To simplify the notation, define

$$T_t = \text{Diag}(t) \text{ and } \Sigma_t = T_t \Sigma T_t + \delta (I - T_t^2).$$
 (7)

Then, our proposed Boolean relaxation of (6) is given by

$$\min_{\boldsymbol{t}\in\mathcal{C}_k} f_{\delta}(\boldsymbol{t}), \quad \text{where} \quad f_{\delta}(\boldsymbol{t}) = -\boldsymbol{t}^{\top} \widetilde{\boldsymbol{\Sigma}}_{\boldsymbol{t}}^{-1} \boldsymbol{t}, \tag{8}$$

and for each k the constraint set C_k is a polytope defined as

$$\mathcal{C}_k = \{ \boldsymbol{t} \in [0, 1]^p : \boldsymbol{t}^\top \boldsymbol{1} \le k \}.$$
(9)

3.2 Theoretical properties

The following result, Theorem 2, shows why (8) is a relaxation of the target problem (6). It shows that $f_{\delta}(t)$ is continuous on the hypercube $[0, 1]^p$ and its shape can be controlled by the auxiliary parameter δ while keeping the values of $f_{\delta}(t)$ fixed—independent of δ —at all the (binary) corners $s \in \{0, 1\}^p$. In addition, (iii) shows that $f_{\delta}(t)$ increases with δ for any fixed interior point t—see Figure 1(a) for an illustration—while (iv) shows that the optimum of (8) is on a simplex.

Theorem 2. The following hold:

- (i) The objective function $f_{\delta}(t)$ in (8) is continuous on $[0, 1]^p$.
- (ii) For every binary vector $s \in \{0,1\}^p$ (i.e., a corner point on the hypercube $[0,1]^p$),

$$f_{\delta}(\boldsymbol{s}) = -\mathbf{1}^{\top} \Sigma_{[\boldsymbol{s}]}^{-1} \mathbf{1}, \text{ for all } \delta > 0.$$

- (iii) For every fixed $t \in (0,1)^p$, $f_{\delta}(t)$ is monotonically increasing in $\delta > 0$.
- (iv) For any $k = 1, \ldots, p$ and $\delta > 0$,

$$\min_{\boldsymbol{t}\in\mathcal{C}_{\boldsymbol{k}}}f_{\delta}(\boldsymbol{t})=\min_{\boldsymbol{t}\in\mathcal{S}_{\boldsymbol{k}}}f_{\delta}(\boldsymbol{t}),$$

here, the simplex $S_k = \{ t \in [0, 1]^p : t^\top \mathbf{1} = k \}$ corresponds to the polytope C_k given in (9).



Figure 1: Panel (a): Surface plots of the auxiliary objective function $f_{\delta}(t)$ for a 2 × 2-dimensional covariance matrix Σ for different values of δ , displaying convexity for $\delta = 0.005$ and concavity for $\delta = 5$. The values of the function at the corners $s \in \{0, 1\}^2$ correspond to those of the discrete function $-\mathbf{1}^{\top} \Sigma_{[s]}^{-1} \mathbf{1}$ in (6). Panel (b): Iterative convergence of t_j 's toward 0 or 1 for a dataset of p = 31 assets when k = 4 during the execution of Algorithm 1. The paths of the four t_j that converge to 1 are shown in black, while other paths are shown in grey.

Our next result, Theorem 3, shows that $f_{\delta}(t)$ exhibits convexity for sufficiently small values of δ and concavity for large ones. Using a 2 × 2 covariance matrix, Figure 1(a) illustrates the convexity of $f_{\delta}(t)$ for small values of δ and concavity for large values of δ , and non-convexity and non-concavity in-between. The proof of this result is based on Lemma 3 in Appendix B, which derives the Hessian of $f_{\delta}(t)$.

Theorem 3. Let η_1 and η_p be the largest and smallest eigenvalues of Σ , respectively. Then,

- (i) $f_{\delta}(t)$ strictly concave over $[0,1]^p$ for $\delta \geq \eta_1$, and
- (ii) for any $\varepsilon \in (0, 1)$, $f_{\delta}(t)$ is strictly convex over $[\varepsilon, 1]^p$ for $\delta \leq 3\eta_p \varepsilon^2 / (1 + 3\varepsilon^2)$.

Strict concavity of $f_{\delta}(t)$ establishes the equivalence between the target problem (6) and the Boolean relaxation (8) for all $\delta \ge \eta_1$. Since (6) is equivalent to the original problem (2), the following result immediately follows from Theorem 3(i).

Corollary 1. There exists $\delta_c \leq \eta_1$ such that for all $\delta \geq \delta_c$, the Boolean relaxation (8) problem is equivalent to the original sparse portfolio selection problem (2).

Finally, Theorem 4 establishes the continuity of the solution to (8) as a function of the auxiliary variable δ . This is a consequence of the well-known optimization result popularly known as *Berge's maximum theorem*; refer to, e.g., Sundaram (1996).

Theorem 4. For each k, suppose h_k^* and \mathcal{D}_k^* are the optimal value and optimal solution of the target problem (6). Further, let

$$h_k(\delta) = \min_{\boldsymbol{t} \in \mathcal{C}_k} f_{\delta}(\boldsymbol{t}) \quad and \quad \mathcal{D}_{k,\delta} = \operatorname*{argmin}_{\boldsymbol{t} \in \mathcal{C}_k} f_{\delta}(\boldsymbol{t}).$$

Then, for any $\epsilon \in (0, \eta_1)$, $h_k(\delta)$ is continuous and $\mathcal{D}_{k,\delta}$ is compact-valued and upper hemicontinuous in $\delta \in [\epsilon, \eta_1]$: for any sequence $\delta_\ell \to \eta_1$ and $\mathbf{t}^{(\ell)} \in \mathcal{D}_{k,\delta}$ with $\mathbf{t}^{(\ell)} \to \mathbf{t}^*$, we have $\mathbf{t}^* \in \mathcal{D}_k^*$.

Since \mathcal{D}_k^* constitutes an optimal solution to the original sparse portfolio problem, Theorem 4 implies that by constructing a sequence of minimizers of $f_{\delta}(t)$ along an increasing sequence of δ values approaching the largest eigenvalue η_1 of Σ , the limit of these minimizers will yield a solution to the original problem. This connection establishes a pathway to recover the sparse portfolio solution through a controlled tuning parameter δ converging to η_1 , inspiring our algorithm below.

3.3 Algorithm

It is well-established that minimizing a strictly convex function over a convex set—such as the polytope C_k or the simplex S_k —is a tractable convex optimization problem, guaranteeing convergence to a unique optimum and enabling ϵ -approximate solutions via gradient-based methods in polynomial time (Boyd and Vandenberghe, 2004). In contrast, minimizing a strictly concave function over the same set is computationally hard, as all optima are confined to the vertices, making gradient descent highly sensitive to initialization and prone to suboptimal solutions. In our context, Theorem 3(i) establishes that $f_{\delta}(t)$ is strictly concave on C_k for any $\delta \geq \eta_1$, leading to an equivalence between the original sparse portfolio selection problem and its reformulation, but limiting the effectiveness of gradient-based methods due to the combinatorial landscape. Conversely, Theorem 3(ii) shows that for sufficiently small positive δ values, $f_{\delta}(t)$ becomes strictly convex over a truncated hypercube $[\varepsilon, 1]^p$, for a given $\varepsilon > 0$, ensuring that gradient-based algorithms reliably converge to the unique optimal solution from any initialization within this region. Building on these properties, our approach begins with a small value of δ to obtain a stable initialization and gradually increases δ toward η_1 , thereby transforming the objective from convex to concave and ultimately guiding the solution to the optimal point of the original binary-constrained problem.

Furthermore, Theorem 2(iv) establishes that the target optimization problem can be restricted to the simplex S_k for each k. The conditional gradient method, also known as the *Frank–Wolfe* algorithm (Frank and Wolfe, 1956; Jaggi, 2013), is particularly well-suited for optimization over simplices and enjoys strong convergence guarantees. Building on this, we introduce a modified variant, Grid-FW, which iteratively updates the vector t while progressively increasing δ from a small initial value up to the largest eigenvalue η_1 of Σ . As mentioned earlier, this continuation strategy is motivated by Theorem 4, which ensures that the optimal solution depends continuously on δ . The full procedure is outlined in Algorithm 1.

Algorithm 1 Grid-FW $(\Sigma, \alpha, \varepsilon, n, m)$

1: Compute the largest and smallest eigenvalues η_1 and η_p of Σ 2: Take $\delta_1 = 3\eta_p \varepsilon^2 / (1 + 3\varepsilon^2)$ and $r = (\eta_1 / \delta_1)^{1/(n-1)}$ 3: Create a geometric grid $\{\delta_1, \delta_2, \dots, \delta_n\}$ where $\delta_\ell = \delta_1 r^{\ell-1}, \ell = 1, \dots, n$ 4: $t \leftarrow (k/p) \cdot \mathbf{1}$ and $s^* \leftarrow \mathbf{0}$ 5: for $\ell = 1, 2, ..., n$ do for i = 1, ..., m do 6: 7: Compute the gradient $\nabla f_{\delta_{\ell}}(t)$ Let $s \in \{0,1\}^p$ with ones at the positions of the k smallest components of $\nabla f_{\delta_\ell}(t)$ 8: 9: $\boldsymbol{t} \leftarrow (1 - \alpha)\boldsymbol{t} + \alpha \boldsymbol{s}$ if $\mathbf{1}_{s^* \leftarrow s}^{\top \Sigma_{[s^*]}} \mathbf{1} < \mathbf{1}^{\top} \Sigma_{[s]} \mathbf{1}$ then 10: 11: 12: end if 13: end for 14: end for 15: return s^* , s and t

Algorithm 1, Grid-FW($\Sigma, \alpha, \varepsilon, n, m$), executes for n epochs and in each epoch, there are m Frank-Wolfe steps. Using Theorem 3, we select $\delta_1 = 3\eta_p \varepsilon^2/(1+3\varepsilon^2)$ for a small positive constant $\varepsilon \leq 0.1(k/p)$, to make sure that the center (k/p)1 of the simplex S_k is within the truncated hypercube $[\varepsilon, 1]^p$. We create an increasing (geometric) sequence $\{\delta_1, \delta_2, \ldots, \delta_n\}$ of n values for the δ parameter with $\delta_\ell = \delta_1 r^{\ell-1}$ and $r = (\eta_1/\delta_1)^{1/n-1}$. Within each epoch ℓ (within the inner for-loop of Algorithm 1), t is iteratively updated m times by solving

$$s \leftarrow \operatorname*{argmin}_{\boldsymbol{u} \in \mathcal{S}_k} \boldsymbol{u}^\top \nabla f_{\delta_\ell}(\boldsymbol{t}), \quad \text{and then,} \quad \boldsymbol{t} \leftarrow (1-\alpha)\boldsymbol{t} + \alpha \boldsymbol{s},$$
(10)

where $\alpha \in (0, 1)$ is a fixed constant. Note that in our problem, it is easy to solve the minimization in (10) because its solution *s* must be a binary vector with ones correspond to the *k* smallest values of the gradient $\nabla f_{\delta_{\ell}}(t)$; see line 8 of the algorithm.

At $\delta_n = \eta_1$, the function $f_{\delta_n}(t)$ becomes strictly concave, so it attains its minimum at a corner of S_k , which is the *final* s returned by line 15. For some datasets, we may slightly increase the performance by considering the *best* model s^* among all models visited by the algorithm (see lines 10-12). Table 1 presents the results for both the final model s and the best model s^* for two real-world datasets. Note that the final and the best models in each case achieve optimal or nearly optimal variance (just one false positive² in two cases). We obtain the optimal models using the Big-M formulation in CPLEX (here p and k are sufficiently small for CPLEX to confirm an optimal solution within a reasonable amount of time). Figure 1(b) shows the convergence of $t^{(\ell)}$ iteratively to a binary point for the dataset with p = 31 when k = 4. We see that $t^{(\ell)}$ converges much before the final epoch n = 500 as indicated by Corollary 1. To take advantage of this, in our implementation of the algorithm, to reduce the running time, we also have an additional termination condition (not stated in the algorithm) that stops the algorithm when t converges close to a corner point.

				1 /						
	Dataset with $p = 31$					Dataset with $p = 85$				
		Final Model		Best Model			Final Model		Best Model	
k	Optimal Variance	FP	% Err	FP	% Err	Optimal Variance	FP	% Err	FP	% Err
1	1.29×10^{-3}	0	0.0	0	0.0	4.50×10^{-4}	0	0.0	0	0.0
2	7.99×10^{-4}	0	0.0	0	0.0	2.74×10^{-4}	0	0.0	0	0.0
3	7.15×10^{-4}	0	0.0	0	0.0	2.19×10^{-4}	0	0.0	0	0.0
4	6.75×10^{-4}	0	0.0	0	0.0	1.97×10^{-4}	1	0.79	1	0.79
5	6.50×10^{-4}	0	0.0	0	0.0	1.84×10^{-4}	0	0.0	0	0.0
6	6.22×10^{-4}	0	0.0	0	0.0	1.72×10^{-4}	0	0.0	0	0.0
7	6.01×10^{-4}	0	0.0	0	0.0	1.64×10^{-4}	0	0.0	0	0.0
8	5.83×10^{-4}	1	0.26	1	0.26	1.56×10^{-4}	0	0.0	0	0.0
9	5.66×10^{-4}	0	0.0	0	0.0	1.52×10^{-4}	0	0.0	0	0.0
10	5.56×10^{-4}	0	0.0	0	0.0	1.47×10^{-4}	0	0.0	0	0.0

Table 1: Illustration of the accuracy of our method compared to optimal models on two datasets containing, respectively, 31 and 85 assets from Chang et al. (2000) for k up to 10. Here, **FP** denotes the number of false positives and **% Err** denotes the percentage of error from our model compared to the optimal variance (i.e., 100 times the ratio of error to the optimal variance).

Remark 1. (*Scaling*) In real-world datasets, the eigenvalues of the covariance matrix Σ can be quite small and span a wide range, as those for real datasets in Table 1 where, for p = 85, the largest and smallest eigenvalues are about 0.024 and 8.18×10^{-5} , respectively, and for p = 31, they are 0.036 and 2.26×10^{-4} . Such small values can cause numerical instability when choosing δ . To address this, we modify the constraint $\mathbf{1}^{\top}\boldsymbol{\beta} = 1$ and replace Σ in (2) with its correlation matrix prior to Boolean relaxation. We incorporate this scaling in our algorithm; see Appendix A for details. Accordingly, η_1 and η_p in Theorems 3 and 4, as well as in Algorithm 1, are rescaled by $\min_j \Sigma_{j,j}$ and $\max_j \Sigma_{j,j}$. For instance, with p = 85, the variances range from 4.49×10^{-4} to 4.62×10^{-3} , so this scaling ensures the δ -grid is numerically stable.

Remark 2. (*Running time*) In Algorithm 1, we perform nm iterations; each iteration requires computing the gradient $\nabla f_{\delta}(t)$ at the current point t. From Lemma 3 in Appendix B, the main cost in

²Here, a false positive refers to our method selecting a suboptimal asset.

each gradient computation stems from solving a linear system $\Pi^{-1}\mathbf{1}$ for a positive definite matrix Π . We do this via conjugate gradient, which takes $O(p^2)$ time within the required accuracy; refer to Golub and Van Loan (1996) or Saad (2003) for details on the complexity of conjugate gradient methods. Consequently, the total runtime of our method is $O(nmp^2)$.

4 Applications

We benchmark our method Grid-FW, Algorithm 1, against the Big-M formulation implemented using CPLEX for three different examples involving real as well as simulated datasets. We set M = 1 for the Big-M formulation, following Bertsimas and Cory-Wright (2022). In Algorithm 1, we set $\alpha = 0.05$, $\varepsilon = \min(0.1k/p, 0.001)$, n = 500, and m = 10. We compare the methods for fixed computational budgets and $k = \lfloor z\% p \rfloor$, z = 10, 25, 50. All experiments were run on a MacBook Pro with an Apple M4 Pro chip and 24 GB of memory, forcing single-threaded execution for numerical libraries (i.e., no parallel computations). Note that, because CPLEX is implemented in highly optimized low-level languages, our Python-based implementation is inherently at a disadvantage. Datasets, Python code for our method, and Julia code for CPLEX are provided in https://github.com/saratmoka/grid-fw.

In Table 2, the first computational budget B (in seconds) is set equal to the time our method takes to solve each problem. To see if CPLEX can improve the solutions with more time, we also consider 60 and 300 seconds for CPLEX, where 300 seconds is the time limit imposed on CPLEX Big-M method by Bertsimas and Cory-Wright (2022) for solving a similar problem. In Example 3, which involves a larger p, CPLEX is given 600 seconds. Across all examples, we compare the variance of the resulting portfolios under each computational budget.

4.1 Example 1: Three portfolio datasets

For our first example, we use three of the five portfolio problems from Chang et al. (2000). Some results for the remaining two smallest datasets, with p = 31,85, are already illustrated in Table 1. Here, we present results for the remaining three datasets with p = 89,98,225.

4.2 Example 2: S&P 500

Our second example involves a dataset from Asimit et al. (2025) containing 6037 daily returns of p = 441 assets from firms that remained continuously listed in the S&P 500 during the period January 2000 to December 2023. We split the dataset into three periods: 2000-2007, 2008-2015, and 2016-2023. Shrinkage is essential when estimating the covariance matrix for portfolio optimization (Ledoit and Wolf, 2004), and we use the analytical non-linear shrinkage estimator in Ledoit and Wolf (2020), which guarantees positive definiteness (recall Assumption 1).

4.3 Example 3: Assets with a low-rank plus noise structure

Our final example consists of simulated data to compare the methods for very large values of p. A common approach (see e.g. Ross, 1976) to model the covariance matrix Σ of the asset return vector \boldsymbol{x}_{τ} is to assume the returns are driven by a linear combination $\Lambda \boldsymbol{\xi}_{\tau}$ of uncorrelated lower dimensional factors $\boldsymbol{\xi}_{\tau} = (\xi_{\tau,1}, \dots, \xi_{\tau,q})^{\top} \in \mathbb{R}^{q}$, where $\Lambda \in \mathbb{R}^{p \times q}$ are the factor loadings. We assume that

$$oldsymbol{x}_{ au} = \Lambda oldsymbol{\xi}_{ au} + oldsymbol{e}, \ oldsymbol{\xi}_{ au} \sim \mathcal{N}(oldsymbol{0}, \mathrm{diag}(
u_1^2, \dots,
u_q^2)), \ oldsymbol{e} \sim \mathcal{N}(oldsymbol{0}, d^2I), \
u_1, \dots,
u_q, d > 0.$$

The resulting covariance is $\Sigma = \Lambda \operatorname{diag}(\nu_1, \ldots, \nu_q) \Lambda^\top + d^2 I$. We divide the assets into two equally large groups, i.e., p/2 assets in each. For the second group, the factor loadings for half of the factors (i.e., q/2), are set to zero. Moreover, the factors in the separate groups are normalized so that the relative variance of the portfolio based on assets from group 1 is roughly 10 times larger than that of group 2. We set q = 1% p and $d^2 = 0.05$, and ν_i^2 simulated uniformly in $[0.02^2, 0.05^2]$.

4.4 Results

Table 2 summarizes the results. Our method finished in at most 560 seconds when p = 3000, and it was often orders of magnitude faster across all examples. CPLEX timed out (over one day) on most

problems, making it impossible to verify optimality. For small-scale problems, our variance reduction over CPLEX with equal run time was modest with only a few percentage points. When CPLEX is allowed more time for small-scales examples, it often finds a better optimum than our method. By contrast, for large-scale problems, our method delivered significantly lower variance than CPLEX with the same budget, and even extra time for CPLEX did not guarantee closing the gap.

Table 2: Results for the three examples in Section 4. **Grid-FW** denotes our method, and the corresponding columns contain the time *B* to run our method with n = 500 and m = 10. For **CPLEX Big-M**, **RV**_B denotes the relative variance difference to **Grid-FW** obtained after *B* seconds. **RV**_{τ} denotes the same quantity obtained after τ seconds. **RV**_{τ} is NA whenever $B > \tau$. A positive value indicates a larger optimal variance (worse solution) for CPLEX. The * symbol denotes a difference of exactly zero.

			Grid-FW		CI	CPLEX Big-M			
Example 1	xample 1 p k		Time B	Var	\mathbf{RV}_B	RV 60	RV 300		
	89	8	0.5	2.11×10^{-4}	$0\%^{*}$	$0\%^*$	$0\%^*$		
		22	0.4	1.61×10^{-4}	0.4%	-0.1%	-0.1%		
		44	0.4	1.39×10^{-4}	1.34%	$0\%^*$	$0\%^*$		
	98	9	0.7	1.30×10^{-4}	4.6%	$0\%^*$	$0\%^*$		
		24	0.6	9.52×10^{-5}	2.5%	$0\%^*$	$0\%^*$		
		49	0.6	8.18×10^{-5}	0.6%	-0.1%	-0.1%		
	225	22	1.5	1.42×10^{-4}	6.64%	-2.7%	-2.5%		
		56	1.6	7.81×10^{-5}	31.3%	-0.7%	-1.7%		
		112	1.5	4.54×10^{-5}	37.7%	3.1%	2.6%		
Example 2	p	k	Time B	Var	\mathbf{RV}_B	RV 60	RV 300		
2000-2007	441	44	3.9	2.95×10^{-5}	21.1%	2.6%	0.6%		
		110	3.8	2.31×10^{-5}	19.5%	3.7%	1.8%		
		220	3.8	2.00×10^{-5}	32.3%	6.9%	0.9%		
2008-2015	441	44	3.7	3.52×10^{-5}	19.5%	3.7%	1.8%		
		110	3.6	2.76×10^{-5}	15.7%	4.7%	1.7%		
		220	3.6	2.43×10^{-5}	39.8%	6.2%	0.7%		
2016-2023	441	44	3.4	4.27×10^{-5}	24.1%	5.5%	1.0%		
		110	3.4	3.42×10^{-5}	23.8%	4.9%	2.0%		
		220	3.4	2.99×10^{-5}	35.6%	4.8%	0.7%		
Example 3	p	k	Time B	Var	\mathbf{RV}_B	RV ₃₀₀	RV 600		
	1000	100	13.6	5.01×10^{-4}	61.3%	-0.1%	-0.1%		
		250	16.0	2.00×10^{-4}	304.3%	0.0%	-0.0%		
		500	20.6	1.00×10^{-4}	708.8%	0.0%	-0.0%		
	2000	200	155.0	2.51×10^{-4}	47.4%	47.4%	0.9%		
		500	168.2	1.00×10^{-4}	269.6%	30.4%	34.4%		
		1000	107.1	5.00×10^{-5}	7.6%	7.6%	1.6%		
	3000	300	558.7	1.68×10^{-4}	28.1%	NA	18.2%		
		750	548.7	6.67×10^{-5}	61.5%	NA	61.5%		
		1500	376.5	3.33×10^{-5}	6.9%	NA	6.9%		

5 Conclusion and future research

We proposed a novel framework for sparse portfolio selection that employs Boolean relaxation to avoid intractable combinatorial computations. We devised an auxiliary objective function with a tuning parameter that transmutes the function from convex to concave. We proved that this transition helps in converging to the optimum of the original problem. Our method—suitable with or without ridge regularization—offers a practical and scalable alternative to the Big-M formulation in CPLEX for sparse portfolio selection.

Future research will consider more general objectives, such as the mean-variance framework, and additional constraints such as no short-selling or minimum and maximum investment limits. We believe such extensions can be competitive alternatives to Bertsimas and Cory-Wright (2022).

A Scaling

To explain our scaling approach, let R be the correlation matrix obtained from the covariance matrix Σ , that is, $R_{i,j} = \sum_{i,j} / \sqrt{\sum_{i,i} \sum_{j,j}}$, $i, j \in \{1, \ldots, p\}$. Let $w = (1/\sqrt{\sum_{1,1}, \ldots, 1/\sqrt{\sum_{p,p}}})$. Then, the binary constrained problem (6) can be rewritten as

$$\min_{\boldsymbol{s}\in\{0,1\}^p} -\boldsymbol{w}_{[\boldsymbol{s}]}^\top R_{[\boldsymbol{s}]}^{-1} \boldsymbol{w}_{[\boldsymbol{s}]}, \quad \text{subject to } |\boldsymbol{s}| \le k.$$
(11)

With $\widetilde{R}_t = T_t R T_t + \delta (I - T_t^2)$ and $\widehat{f}_{\delta}(t) = -(\boldsymbol{w} \odot \boldsymbol{t})^\top \widetilde{R}_t^{-1}(\boldsymbol{w} \odot \boldsymbol{t}), \boldsymbol{t} \in [0, 1]^p$, we can consider a Boolean relaxation of (11) as

$$\min_{\boldsymbol{t}\in\mathcal{C}_k}\widehat{f}_{\delta}(\boldsymbol{t}),$$

where C_k is the polytope defined by (9). In particular, at the interior points $t \in (0,1)^p$, we can simplify the objective function to be

$$\widehat{f}_{\delta}(t) = -\mathbf{1}^{\top} \widehat{\Pi}_{t}^{-1} \mathbf{1}, \text{ where } \widehat{\Pi}_{t} = \Sigma + \delta D_{t} \operatorname{Diag}(v),$$

with $v = 1/(w \odot w)$, which is the diagonal of Σ (i.e., variances of the assets). See Lemma 3 in Appendix B for expressions of the gradient and the Hessian of $\widehat{f}_{\delta}(t)$ at each $t \in (0, 1)^p$. Furthermore, the proof of Theorem 3 in Appendix B establishes that

- (i) $\widehat{f}_{\delta}(t)$ strictly concave over $[0,1]^p$ for $\delta \geq \eta_1 / \min_j v_j$, and
- (ii) for every $\varepsilon \in (0, 1)$, it is strictly convex over $[\varepsilon, 1]^p$ for $\delta \leq \frac{\eta_p}{\max_j v_j} \left(\frac{3\varepsilon^2}{1+3\varepsilon^2}\right)$.

Here, recall that η_1 and η_p are the largest and smallest eigenvalues of Σ , respectively.

B Proofs

Proof of Lemma 1. The problem is convex and has a unique minimum. Using the method of Lagrange multipliers gives the result. \Box

Proof of Theorem 1. It is well-known that every principal submatrix of a symmetric positive definite matrix is also positive definite (Golub and Van Loan, 1996). Since Σ is symmetric positive definite, this implies that $\Sigma_{[s]}$ is also positive definite for every binary vector $s \neq 0$, and hence

$$h(k) \coloneqq \max_{\boldsymbol{s} \in \{0,1\}^p, \ \boldsymbol{1}^\top \boldsymbol{s} = k} \boldsymbol{1}^\top \boldsymbol{\Sigma}_{[\boldsymbol{s}]}^{-1} \boldsymbol{1}$$

is well-defined for all $k = 1, \ldots, p$.

Now fix k and consider $s \in \{0, 1\}^p$ such that $\mathbf{1}^{\top} s = k$. Consider s' with $\mathbf{1}^{\top} s' = k + 1$ such that the Hamming distance between s and s' is 1, that is, there is a unique $j \in \{1, \ldots, p\}$ such that $s_i = s'_i$ for all $i \neq j$ and $0 = s_j < s'_j = 1$. Then, the proof is complete if we show that

$$\mathbf{1}^{\top} \Sigma_{[\boldsymbol{s}]}^{-1} \mathbf{1} \le \mathbf{1}^{\top} \Sigma_{[\boldsymbol{s}']}^{-1} \mathbf{1},$$
(12)

because, under (12), $\mathbf{1}^{\top} \Sigma_{[s]}^{-1} \mathbf{1} \leq h(k+1)$ and hence $h(k) \leq h(k+1)$ by maximizing the left hand size over all s with |s| = k.

We prove (12) using the popular Banachiewicz inversion lemma (Tian and Takane, 2005). Without loss of generality we assume that such s and s' differ at the index p. Then, we write

$$\Sigma_{[s']} = \begin{bmatrix} \Sigma_{[s]} & a \\ & & \\ a^{\top} & b \end{bmatrix},$$

for some appropriate vector \boldsymbol{a} and a scalar b. Note that both $\Sigma_{[\boldsymbol{s}']}$ and $\Sigma_{[\boldsymbol{s}]}$ are invertible and b > 0. Therefore, the Schur complement $c = b - \boldsymbol{a}^{\top} \Sigma_{[\boldsymbol{s}]}^{\top} \boldsymbol{a}$ must be non-zero; see Tian and Takane (2005). Then,

$$\Sigma_{[s']}^{-1} = \begin{bmatrix} I & -\frac{1}{c} \Sigma_{[s]}^{-1} a \\ 0^{\top} & \frac{1}{c} \end{bmatrix} \begin{bmatrix} \Sigma_{[s]}^{-1} & 0 \\ -a^{\top} \Sigma_{[s]}^{-1} & 1 \end{bmatrix} = \begin{bmatrix} \Sigma_{[s]}^{-1} + \frac{1}{c} \Sigma_{[s]}^{-1} a a^{\top} \Sigma_{[s]}^{-1} & -\frac{1}{c} \Sigma_{[s]}^{-1} a \\ -\frac{1}{c} a^{\top} \Sigma_{[s]}^{-1} & \frac{1}{c} \end{bmatrix}.$$

Note that for any matrix A, $\mathbf{1}^{\top}A\mathbf{1}$ is simply the sum of all the elements of A. Similarly, for any vector $\boldsymbol{u}, \mathbf{1}^{\top}\boldsymbol{u}$ is the sum of all the elements of \boldsymbol{u} . Thus,

$$\mathbf{1}^{\top} \Sigma_{[s']}^{-1} \mathbf{1} = \mathbf{1}^{\top} \Sigma_{[s]}^{-1} \mathbf{1} + \frac{1}{c} (\mathbf{1}^{\top} \Sigma_{[s]}^{-1} a)^2 - 2 \frac{1}{c} \mathbf{1}^{\top} \Sigma_{[s]}^{-1} a + \frac{1}{c} = \mathbf{1}^{\top} \Sigma_{[s]}^{-1} \mathbf{1} + \frac{1}{c} \left(\mathbf{1}^{\top} \Sigma_{[s]}^{-1} a - 1 \right)^2.$$

Therefore, to establish (12), we just need to show that c > 0. For this, using simple block-matrix multiplication, observe that

$$\begin{bmatrix} I & \mathbf{0} \\ -\mathbf{a}^{\top} \Sigma_{[\mathbf{s}]}^{-1} & 1 \end{bmatrix} \begin{bmatrix} \Sigma_{[\mathbf{s}]} & \mathbf{a} \\ \mathbf{a}^{\top} & b \end{bmatrix} \begin{bmatrix} I & -\Sigma_{[\mathbf{s}]}^{-1} \mathbf{a} \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} \Sigma_{[\mathbf{s}]} & \mathbf{0} \\ \mathbf{0}^{\top} & c \end{bmatrix}$$

Since $\Sigma_{[s']}$ is positive definite, for any scalar $v \neq 0$ with $\boldsymbol{u} = [\boldsymbol{0}^{\top}, v]^{\top}$, using the left hand side expression of the above equality,

$$\boldsymbol{u}^{\top} \begin{bmatrix} I & \boldsymbol{0} \\ -\boldsymbol{a}^{\top} \boldsymbol{\Sigma}_{[\boldsymbol{s}]}^{-1} & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{\Sigma}_{[\boldsymbol{s}]} & \boldsymbol{a} \\ \boldsymbol{a}^{\top} & \boldsymbol{b} \end{bmatrix} \begin{bmatrix} I & -\boldsymbol{\Sigma}_{[\boldsymbol{s}]}^{-1} \boldsymbol{a} \\ \boldsymbol{0} & 1 \end{bmatrix} \boldsymbol{u} = \begin{bmatrix} -v\boldsymbol{a}^{\top} \boldsymbol{\Sigma}_{[\boldsymbol{s}]}^{-1} & v \end{bmatrix} \boldsymbol{\Sigma}_{[\boldsymbol{s}']} \begin{bmatrix} -v\boldsymbol{\Sigma}_{[\boldsymbol{s}]}^{-1} \boldsymbol{a} \\ v \end{bmatrix} > 0,$$
and hence

and hence,

$$\boldsymbol{u}^{\top} \begin{bmatrix} \Sigma_{[\boldsymbol{s}]} & \boldsymbol{0} \\ \boldsymbol{0}^{\top} & c \end{bmatrix} \boldsymbol{u} = cv^2 > 0,$$

which is possible only if c > 0.

We need the following lemma in proving Theorem 2(iii).

Lemma 2. For every interior point $t \in (0,1)^p$, the objective function $f_{\delta}(t)$ can be expressed as

$$f_{\delta}(t) = -\mathbf{1}^{\top} \Pi_{t}^{-1} \mathbf{1}, \text{ with } \Pi_{t} = \Sigma + \delta D_{t} \text{ and } D_{t} = T_{t}^{-2} - I$$

Proof. Note that $\mathbf{t}^{\top} = \mathbf{1}^{\top} T_t$. Then,

$$f_{\delta}(t) = -\mathbf{1}^{\top} T_{t} \Sigma_{t}^{-1} T_{t} \mathbf{1}$$

= $-\mathbf{1}^{\top} \left(T_{t}^{-1} \widetilde{\Sigma}_{t} T_{t}^{-1} \right)^{-1} \mathbf{1}$
= $-\mathbf{1}^{\top} \left(\Sigma + \delta (T_{t}^{-2} - I) \right)^{-1} \mathbf{1}$
= $-\mathbf{1}^{\top} \Pi_{t}^{-1} \mathbf{1},$

since T_t is invertible for every interior point t.

Proof of Theorem 2. To establish the continuity in (i), let $t_1, t_2, \dots \in [0, 1]^p$ be a sequence of points converging to t (element-wise or in the Euclidean norm). If t contains zeros, without loss of generality, assume all these zeros appear at the front. Then, we can write

$$T_{\boldsymbol{t}_{\ell}} = \begin{bmatrix} F'_{\boldsymbol{t}_{\ell}} & \boldsymbol{0} \\ \boldsymbol{0} & F''_{\boldsymbol{t}_{\ell}} \end{bmatrix},$$

where $F'_{t_{\ell}}$ is a diagonal matrix of the front elements of t_{ℓ} (that converge to zero) and $F''_{t_{\ell}}$ is a diagonal matrix of the remaining elements. Therefore, for all sufficiently large ℓ , we can assume that $F''_{t_{\ell}}$ is invertible. Recall from (7) that $\tilde{\Sigma}_t = T_t \Sigma T_t + \delta(I - T_t^2)$. By dividing the Σ matrix into four suitable sub-matrices, we can write

$$\begin{split} \widetilde{\Sigma}_{t_{\ell}} &= \begin{bmatrix} F_{t_{\ell}}' & \mathbf{0} \\ \mathbf{0} & F_{t_{\ell}}'' \end{bmatrix} \begin{bmatrix} \Sigma' & \overline{\Sigma} \\ \overline{\Sigma}^{\top} & \Sigma'' \end{bmatrix} \begin{bmatrix} F_{t_{\ell}}' & \mathbf{0} \\ \mathbf{0} & F_{t_{\ell}}'' \end{bmatrix} + \delta \begin{bmatrix} I - (F_{t_{\ell}}')^2 & \mathbf{0} \\ \mathbf{0} & I - (F_{t_{\ell}}'')^2 \end{bmatrix} \\ &= \begin{bmatrix} F_{t_{\ell}}' \Sigma^{\top} F_{t_{\ell}}' & F_{t_{\ell}}' \overline{\Sigma} F_{t_{\ell}}'' \\ F_{t_{\ell}}' \overline{\Sigma}^{\top} F_{t_{\ell}}' & F_{t_{\ell}}' \Sigma'' F_{t_{\ell}}'' \end{bmatrix} + \delta \begin{bmatrix} I - (F_{t_{\ell}}')^2 & \mathbf{0} \\ \mathbf{0} & I - (F_{t_{\ell}}'')^2 \end{bmatrix} \\ &= \underbrace{\begin{bmatrix} \delta \left(I - (F_{t}')^2 \right) & \mathbf{0} \\ \mathbf{0} & F_{t}'' \Sigma'' F_{t}'' + \delta \left(I - (F_{t}'')^2 \right) \end{bmatrix}}_{\widetilde{\Sigma}_{t}} \\ &+ \underbrace{\begin{bmatrix} F_{t_{\ell}}' \Sigma' F_{t_{\ell}}' & F_{t_{\ell}}' \overline{\Sigma} F_{t_{\ell}}'' \\ F_{t_{\ell}}' \overline{\Sigma}^{\top} F_{t_{\ell}}' & F_{t_{\ell}}'' \Sigma'' F_{t_{\ell}}'' - F_{t}'' \Sigma'' F_{t}'' \end{bmatrix}}_{E_{\ell}} + \delta \begin{bmatrix} (F_{t}')^2 - (F_{t_{\ell}}')^2 & \mathbf{0} \\ \mathbf{0} & (F_{t}'')^2 - (F_{t_{\ell}}'')^2 \end{bmatrix}}_{E_{\ell}} \end{split}$$

$$=\Sigma_t + E_\ell,$$

where $E_{\ell} = \widetilde{\Sigma}_{t_{\ell}} - \widetilde{\Sigma}_{t}$ clearly converges to an all-zero matrix. That means, the spectral radius of E_{ℓ} is less than 1 for all sufficiently large values of ℓ . Thus using the Neumann series (Horn and Johnson, 2012), we can show that

$$\widetilde{\Sigma}_{\boldsymbol{t}_{\ell}}^{-1} = (\widetilde{\Sigma}_{\boldsymbol{t}} + E_{\ell})^{-1} \to \widetilde{\Sigma}_{\boldsymbol{t}}^{-1}, \quad \text{as } \ell \to \infty.$$

As a consequence, $\lim_{\ell \to \infty} f_{\delta}(t_{\ell}) = f_{\delta}(t)$ for any $\delta > 0$.

To establish (ii), observe that for any binary vector $s \in \{0, 1\}^p$,

$$(\widetilde{\Sigma}_{\boldsymbol{s}})_{[\boldsymbol{s}]} = (T_{\boldsymbol{s}}\Sigma T_{\boldsymbol{s}})_{[\boldsymbol{s}]} = \Sigma_{[\boldsymbol{s}]}, \text{ and } (\widetilde{\Sigma}_{\boldsymbol{s}})_{[\boldsymbol{1}-\boldsymbol{s}]} = \delta I.$$

Thus, $f_{\delta}(\boldsymbol{s}) = -\mathbf{1}^{\top} \Sigma_{[\boldsymbol{s}]}^{-1} \mathbf{1}.$

To establish (iii), observe from Lemma 2 that

$$egin{aligned} &rac{d}{d\delta}f_{\delta}(oldsymbol{t}) = -oldsymbol{1}^{ op}rac{d}{d\delta}\Pi_{oldsymbol{t}}^{-1}oldsymbol{1}^{ op} \ &= oldsymbol{1}^{ op}\Pi_{oldsymbol{t}}^{-1}oldsymbol{1}\left(rac{d}{d\delta}\Pi_{oldsymbol{t}}
ight)\Pi_{oldsymbol{t}}^{-1}oldsymbol{1}\ &= oldsymbol{u}^T\left(rac{d}{d\delta}\Pi_{oldsymbol{t}}
ight)oldsymbol{u}\ &= oldsymbol{u}^T\left(T_{oldsymbol{t}}^{-2}-I
ight)oldsymbol{u}, \end{aligned}$$

with $u = \prod_{t=1}^{t} 1 \neq 0$. Since t is in the unit interval, the matrix in the quadratic form is positive definite, which concludes the claim.

Finally, (iv) is a simple consequence of Lemma 3 below which show that all the elements gradient $\nabla f_{\delta}(t)$ are non-positive. That means, in each coordinate t_j , the objective function $f_{\delta}(t)$ is non-increasing along t_j . Thus, for each k, the simplex S_k contains an optimal solution of Boolean relaxation (8).

We now establish the gradient and the Hessian of the objective function $\hat{f}_{\delta}(t)$ using the scaling in Section A. Without scaling, the gradient and the Hessian of $f_{\delta}(t)$ are obtained by replacing v with 1 in this lemma.

Lemma 3. The gradient and the Hessian of $\widehat{f}_{\delta}(t)$ at every $t \in (0,1)^p$ are respectively given by

$$abla \widehat{f}_{\delta}(\boldsymbol{t}) = -2\delta rac{\boldsymbol{v} \odot (\widehat{\Pi}_{\boldsymbol{t}}^{-1} \mathbf{1})^2}{\boldsymbol{t}^3},$$

and

$$\widehat{H}_{\delta}(\boldsymbol{t}) = 2\delta \operatorname{Diag}\left(\frac{\boldsymbol{v} \odot (\widehat{\Pi}_{\boldsymbol{t}}^{-1} \boldsymbol{1})}{\boldsymbol{t}^{3}}\right) \left[3\operatorname{Diag}(\boldsymbol{t}^{2} \odot \boldsymbol{w}^{2}) - 4\delta \widehat{\Pi}_{\boldsymbol{t}}^{-1}\right] \operatorname{Diag}\left(\frac{\boldsymbol{v} \odot (\widehat{\Pi}_{\boldsymbol{t}}^{-1} \boldsymbol{1})}{\boldsymbol{t}^{3}}\right),$$

where $\widehat{\Pi}_t = \Sigma + \delta D_t \operatorname{Diag}(v)$, and the division and power operations on vectors are element-wise.

Proof. Similar to Lemma 2, we can show that for any interior point $t \in (0, 1)^p$,

$$\widehat{f}_{\delta}(t) = -\mathbf{1}^{\top} \widehat{\Pi}_{t}^{-1} \mathbf{1}$$

Thus, for each $j = 1, \ldots, p$,

$$\frac{\partial \widehat{f}_{\delta}(\boldsymbol{t})}{\partial t_{j}} = -\mathbf{1}^{\top} \frac{\partial \widehat{\Pi}_{\boldsymbol{t}}^{-1}}{\partial t_{j}} \mathbf{1} = \mathbf{1}^{\top} \widehat{\Pi}_{\boldsymbol{t}}^{-1} \frac{\partial \widehat{\Pi}_{\boldsymbol{t}}}{\partial t_{j}} \widehat{\Pi}_{\boldsymbol{t}}^{-1} \mathbf{1}.$$

Since Σ does not depend on t, from the definition of Π_t ,

$$\frac{\partial \widehat{\Pi}_{t}}{\partial t_{j}} = \delta \frac{\partial D_{t}}{\partial t_{j}} \operatorname{Diag}(\boldsymbol{v}) = -\frac{2\delta v_{j}}{t_{j}^{3}} \boldsymbol{e}_{j} \boldsymbol{e}_{j}^{\top},$$
(13)

where e_j is the vector with 1 at the *j*-th location and 0 everywhere else. Thus, we get the required expression for the gradient $\nabla \widehat{f}_{\delta}(t)$.

We now derive an expression of the Hessian of $\widehat{f}(t)$. Toward this, to simplify the notation, let $u_t = \widehat{\Pi}_t^{-1} \mathbf{1}$. Then, *j*-th column of the Hessian is given by

$$\begin{split} \frac{\partial}{\partial t_j} \nabla \widehat{f}_{\delta}(\boldsymbol{t}) &= \frac{6\delta v_j}{t_j^4} \boldsymbol{e}_j \boldsymbol{e}_j^\top (\boldsymbol{u}_{\boldsymbol{t}} \odot \boldsymbol{u}_{\boldsymbol{t}}) - 2\delta \left(\frac{1}{\boldsymbol{t}^3}\right) \odot \frac{\partial}{\partial t_j} (\boldsymbol{u}_{\boldsymbol{t}} \odot \boldsymbol{u}_{\boldsymbol{t}}) \\ &= \frac{6\delta v_j}{t_j^4} \boldsymbol{e}_j \boldsymbol{e}_j^\top (\boldsymbol{u}_{\boldsymbol{t}} \odot \boldsymbol{u}_{\boldsymbol{t}}) - 4\delta \left(\frac{\boldsymbol{u}_{\boldsymbol{t}}}{\boldsymbol{t}^3}\right) \odot \frac{\partial \boldsymbol{u}_{\boldsymbol{t}}}{\partial t_j} \\ &= \frac{6\delta v_j}{t_j^4} \boldsymbol{e}_j \boldsymbol{e}_j^\top (\boldsymbol{u}_{\boldsymbol{t}} \odot \boldsymbol{u}_{\boldsymbol{t}}) + 4\delta \left(\frac{\boldsymbol{u}_{\boldsymbol{t}}}{\boldsymbol{t}^3}\right) \odot \left(\widehat{\Pi}_{\boldsymbol{t}}^{-1} \frac{\partial \widehat{\Pi}_{\boldsymbol{t}}}{\partial t_j} \boldsymbol{u}_{\boldsymbol{t}}\right). \end{split}$$

Using (13), we can write

$$\frac{\partial}{\partial t_j} \nabla \widehat{f}_{\delta}(\boldsymbol{t}) = \frac{6\delta v_j}{t_j^4} \boldsymbol{e}_j \boldsymbol{e}_j^\top (\boldsymbol{u}_{\boldsymbol{t}} \odot \boldsymbol{u}_{\boldsymbol{t}}) - \frac{8\delta^2 v_j}{t_j^3} \left(\frac{\boldsymbol{u}_{\boldsymbol{t}}}{\boldsymbol{t}^3}\right) \odot \left(\widehat{\Pi}_{\boldsymbol{t}}^{-1} \boldsymbol{e}_j \boldsymbol{e}_j^\top \boldsymbol{u}_{\boldsymbol{t}}\right).$$

Consequently, the Hessian of f_{δ} at t is given by

$$\begin{split} \widehat{H}_{\delta}(t) &= 6\delta \left[\operatorname{Diag}\left(\frac{\boldsymbol{v} \odot \boldsymbol{u}_{t}}{t^{3}}\right) \right]^{2} T_{t}^{2} \operatorname{Diag}(1/\boldsymbol{v}) - 8\delta^{2} \operatorname{Diag}\left(\frac{\boldsymbol{v} \odot \boldsymbol{u}_{t}}{t^{3}}\right) \widehat{\Pi}_{t}^{-1} \operatorname{Diag}\left(\frac{\boldsymbol{v} \odot \boldsymbol{u}_{t}}{t^{3}}\right) \\ &= 2\delta \operatorname{Diag}\left(\frac{\boldsymbol{v} \odot \boldsymbol{u}_{t}}{t^{3}}\right) \left[3 \operatorname{Diag}(t^{2} \odot \boldsymbol{w}^{2}) - 4\delta \widehat{\Pi}_{t}^{-1} \right] \operatorname{Diag}\left(\frac{\boldsymbol{v} \odot \boldsymbol{u}_{t}}{t^{3}}\right), \end{split}$$
hich completes the proof by recalling that $\boldsymbol{w} = 1/\sqrt{\boldsymbol{v}}.$

which completes the proof by recalling that $w = 1/\sqrt{v}$.

Proof of Theorem 3. We establish the proof in generality under scaling, and Theorem 3 (no scaling) immediately holds by replacing v with 1.

The Hessian $\hat{H}_{\delta}(t)$ is clearly symmetric at every $t \in (0,1)^p$ and $\delta > 0$, but not necessarily positive definite. To show (i), note from Lemma 3 that $\hat{H}_{\delta}(t)$ is negative definite if $4\delta \hat{\Pi}_{t}^{-1} - 3 \operatorname{Diag}(t^{2} \odot w^{2})$ is positive definite. To simplify the notation, for a symmetric matrix A, we write $A \ge 0$ to denote that A is positive semi-definite.

Suppose that $\Sigma = U\Delta U^{\top}$ is the singular value decomposition of Σ . Then, $\Sigma \leq \eta_1 I$ and thus,

$$\Pi_{\boldsymbol{t}} = \Sigma + \delta D_{\boldsymbol{t}}\operatorname{Diag}(\boldsymbol{v}) \leq \eta_1 I + \delta D_{\boldsymbol{t}}\operatorname{Diag}(\boldsymbol{v}) = \operatorname{Diag}\left(\eta_1 \mathbf{1} + \delta(\mathbf{1}/\boldsymbol{t}^2 - \mathbf{1})\odot\boldsymbol{v}\right).$$

Hence,

$$\widehat{\Pi}_{\boldsymbol{t}}^{-1} \geq \operatorname{Diag}\left(\frac{\boldsymbol{t}^2}{\eta_1 \boldsymbol{t}^2 + \delta(\boldsymbol{1} - \boldsymbol{t}^2) \odot \boldsymbol{v}}\right),$$

and thus,

$$4\delta\widehat{\Pi}_{\boldsymbol{t}}^{-1} - 3\operatorname{Diag}(\boldsymbol{t}^2\odot\boldsymbol{w}^2) \geq \operatorname{Diag}\left(\frac{4\delta\boldsymbol{t}^2}{\eta_1\boldsymbol{t}^2 + \delta(\boldsymbol{1} - \boldsymbol{t}^2)\odot\boldsymbol{v}} - 3\frac{\boldsymbol{t}^2}{\boldsymbol{v}}\right),$$

here we used the fact that $v = 1/w^2$. Now it is enough to show that

$$\frac{4\delta}{\eta_1 t_j^2 + \delta(1 - t_j^2)v_j} - \frac{3}{v_j} \ge 0$$

for all j and $t_j \in (0, 1)$. Or equivalently,

$$4\delta \ge \frac{3}{v_j} \left(\eta_1 t_j^2 + \delta(1 - t_j^2) v_j \right)$$

That is,

$$(1+3t_j^2))\delta \ge 3(\eta_1/v_j)t_j^2,$$

or,

$$\delta - 3(\eta_1/v_j - \delta)t_j^2 \ge 0.$$

Now, to establish the convexity property in (ii), observe that

$$\widehat{\Pi}_{t} = \Sigma + \delta D_{t} \operatorname{Diag}(\boldsymbol{v}) \ge \eta_{p} I + \delta D_{t} \operatorname{Diag}(\boldsymbol{v}) = \operatorname{Diag}\left(\eta_{p} \mathbf{1} + \delta(\mathbf{1}/t^{2} - \mathbf{1}) \odot \boldsymbol{v}\right).$$

e.

Hence,

$$\widehat{\Pi}_{oldsymbol{t}}^{-1} \leq \operatorname{Diag}\left(rac{oldsymbol{t}^2}{\eta_poldsymbol{t}^2 + \delta(oldsymbol{1} - oldsymbol{t}^2) \odot oldsymbol{v}}
ight),$$

and thus,

$$4\delta\widehat{\Pi}_{\boldsymbol{t}}^{-1} - 3\operatorname{Diag}(\boldsymbol{t}^2\odot\boldsymbol{w}^2) \leq \operatorname{Diag}\left(\frac{4\delta\boldsymbol{t}^2}{\eta_p\boldsymbol{t}^2 + \delta(\boldsymbol{1} - \boldsymbol{t}^2)\odot\boldsymbol{v}} - 3\frac{\boldsymbol{t}^2}{\boldsymbol{v}}\right).$$

Now it is enough to show that

$$\frac{4\delta}{\eta_p t_j^2 + \delta(1 - t_j^2)v_j} - \frac{3}{v_j} \le 0$$

for all j. Equivalently, $4\delta v_j \leq 3\eta_p t_j^2 + 3\delta(1-t_j^2)v_j$. That is, $\delta v_j \leq 3\eta_p t_j^2 - 3\delta v_j t_j^2$, or,

$$\delta \le \frac{1}{v_j} \left(\frac{3\eta_p t_j^2}{1 + 3v_j t_j^2} \right)$$

which is true when for all $t \in (\varepsilon, 1)^p$ and $\delta > 0$ such that

$$\delta \leq \frac{1}{\max_j v_j} \left(\frac{3\eta_p \varepsilon^2}{1 + 3\varepsilon^2} \right).$$

Proof of Theorem 4. Recall that for a fixed integer k,

$$h_k(\delta) = \min_{oldsymbol{t}\in\mathcal{C}_k} f_\delta(oldsymbol{t}) \quad ext{and} \quad \mathcal{D}_{k,\delta} = \operatorname*{argmin}_{oldsymbol{t}\in\mathcal{C}_k} f_\delta(oldsymbol{t}).$$

Since $\Delta := [\varepsilon, \eta_1]$ is compact for any $\varepsilon \in (0, \eta_1)$, from Chapter 9 of Sundaram (1996), $h_k(\delta)$ is continuous and $\mathcal{D}_{k,\delta}$ is compact-valued, upper hemicontinuous³ correspondence on Δ . This implies that for any sequence $\delta_\ell \to \eta_1$ and $\mathbf{t}^{(\ell)} \in \mathcal{D}_{k,\delta}$ with $\mathbf{t}^{(\ell)} \to \mathbf{t}^*$, we have $\mathbf{t}^* \in \mathcal{D}_{k,\eta_1}$. We known from Theorem 3(i) that $f_{\eta_1}(\mathbf{t})$ is strictly concave on \mathcal{C}_k , and thus, \mathcal{D}_{k,η_1} must be the solution set \mathcal{D}_k^* of the target optimization problem (6).

³Upper hemicontinuity is sometimes referred to as upper semicontinuity.

References

- Asimit, V., Peng, L., Tunaru, R., and Zhou, F. (2025). Risk budgeting under general risk measures. https://openaccess.city.ac.uk/id/eprint/33733/.
- Bertsimas, D. and Cory-Wright, R. (2022). A scalable algorithm for sparse portfolio selection. *INFORMS Journal on Computing*, 34(3):1489–1511.
- Bertsimas, D., Cory-Wright, R., and Pauphilet, J. (2021). A unified approach to mixed-integer optimization problems with logical constraints. *SIAM Journal on Optimization*, 31(3):2340–2367.
- Bertsimas, D., King, A., and Mazumder, R. (2016). Best subset selection via a modern optimization lens. *The Annals of Statistics*, 44(2):813 852.
- Bhatia, R. (2009). Positive definite matrices. Princeton university press.
- Boyd, S. P. and Vandenberghe, L. (2004). Convex optimization. Cambridge university press.
- Brodie, J., Daubechies, I., De Mol, C., Giannone, D., and Loris, I. (2009). Sparse and stable Markowitz portfolios. *Proceedings of the National Academy of Sciences*, 106(30):12267–12272.
- Chang, T.-J., Meade, N., Beasley, J. E., and Sharaiha, Y. M. (2000). Heuristics for cardinality constrained portfolio optimisation. *Computers & Operations Research*, 27(13):1271–1302.
- Clarke, R., De Silva, H., and Thorley, S. (2011). Minimum-variance portfolio composition. *Journal* of *Portfolio Management*, 37(2):31.
- DeMiguel, V., Garlappi, L., Nogales, F. J., and Uppal, R. (2009). A generalized approach to portfolio optimization: Improving performance by constraining portfolio norms. *Management Science*, 55(5):798–812.
- Fan, J., Zhang, J., and Yu, K. (2012). Vast portfolio selection with gross-exposure constraints. *Journal of the American Statistical Association*, 107(498):592–606.
- Fastrich, B., Paterlini, S., and Winker, P. (2015). Constructing optimal sparse portfolios using regularization methods. *Computational Management Science*, 12(3):417–434.
- Frank, M. and Wolfe, P. (1956). An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3(1-2):95–110.
- Gao, J. and Li, D. (2013). Optimal cardinality constrained portfolio selection. *Operations Research*, 61(3):745–761.
- Golub, G. H. and Van Loan, C. F. (1996). *Matrix Computations*. The Johns Hopkins University Press, third edition.
- Gurobi Optimization, LLC (2024). Gurobi optimizer reference manual.
- Hastie, T., Tibshirani, R., and Wainwright, M. (2015). Statistical learning with sparsity. *Monographs* on Statistics and Applied Probability, 143(143):8.
- Horn, R. A. and Johnson, C. R. (2012). Matrix Analysis. Cambridge University Press.
- IBM Corporation (2024). IBM ILOG CPLEX Optimization Studio.
- Jaggi, M. (2013). Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In International Conference on Machine Learning, pages 427–435. PMLR.
- Land, A. H. and Doig, A. G. (1960). An automatic method of solving discrete programming problems. *Econometrica*, 28(3):497–520.
- Ledoit, O. and Wolf, M. (2004). Honey, I shrunk the sample covariance matrix. *Journal of Portfolio Management*, 30(4):110–119.
- Ledoit, O. and Wolf, M. (2020). Analytical nonlinear shrinkage of large-dimensional covariance matrices. *The Annals of Statistics*, 48(5):3043–3065.

Markowitz, H. (1952). Portfolio selection. The Journal of Finance, 7:77-91.

- Mencarelli, L. and d'Ambrosio, C. (2019). Complex portfolio selection via convex mixed-integer quadratic programming: A survey. *International Transactions in Operational Research*, 26(2):389–414.
- Moka, S., Liquet, B., Zhu, H., and Muller, S. (2024). COMBSS: best subset selection via continuous optimization. *Statistics and Computing*, 34(2):75.
- Natarajan, B. K. (1995). Sparse approximate solutions to linear systems. *SIAM Journal on Computing*, 24(2):227–234.

Roncalli, T. (2013). Introduction to Risk Parity and Budgeting. CRC Press.

Ross, S. A. (1976). The arbitrage theory of capital asset pricing. *Journal of Economic Theory*, 13(3):341–360.

Saad, Y. (2003). Iterative methods for sparse linear systems. SIAM.

- Sundaram, R. K. (1996). A First Course in Optimization Theory. Cambridge University Press.
- Tian, Y. and Takane, Y. (2005). Schur complements and Banachiewicz-Schur forms. *Electronic Journal of Linear Algebra*, 13:405–418.
- Tillmann, A. M., Bienstock, D., Lodi, A., and Schwartz, A. (2024). Cardinality minimization, constraints, and regularization: a survey. *SIAM Review*, 66(3):403–477.